Social Robot

USER'S MANUAL

Kristine Dietrich
Willy Douglas
Laura Hendrickson
Emily Wetmore

May 27, 2013

USER'S MANUAL

TABLE OF CONTENTS

Page #

1.0 GENERAL INFORMATION 1-1

1.1 System Overview 1-1

1.2 Points of Contact 1-1

1.2.1 Information 1-1

2.0 SYSTEM SUMMARY 2-1

2.1 System Configuration 2-1

2.2 Function Flows 2-2

3.0 GETTING STARTED 3-1

3.1 Downloading the Social Robot Program 3-1

3.2 About the Electrical Box 3-11

3.3 Using the GUI 3-12

3.4 Advanced GUI Settings 3-13

4.0 TROUBLESHOOTING 4-17

4.1 Common Problems 4-17

4.2 Using the Debug Menu 4-19

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

1.1 System Overview

The purpose of this project was to create a Socially Assistive Robot that will aid children with cognitive disorders in learning the basic ability of "reach." This reaching milestone is a precursor to many other important milestones, namely a back-and-forth conversation, something with which many children with cognitive disorders struggle. The team created a robot-Kinect interface that sees the user and mimics their motions. The Kinect is an image-based movement sensor that sees four unique arm movements of the user. It sends this information through a microprocessor to the robot in order for the robot to mimic the user's actions. This document is intended for Mr. Melonis and the Assistive Technology Partners. It is also intended for the members of Engineering Design Class, the instructors, and the faculty consultants for this project.

1.2 Points of Contact

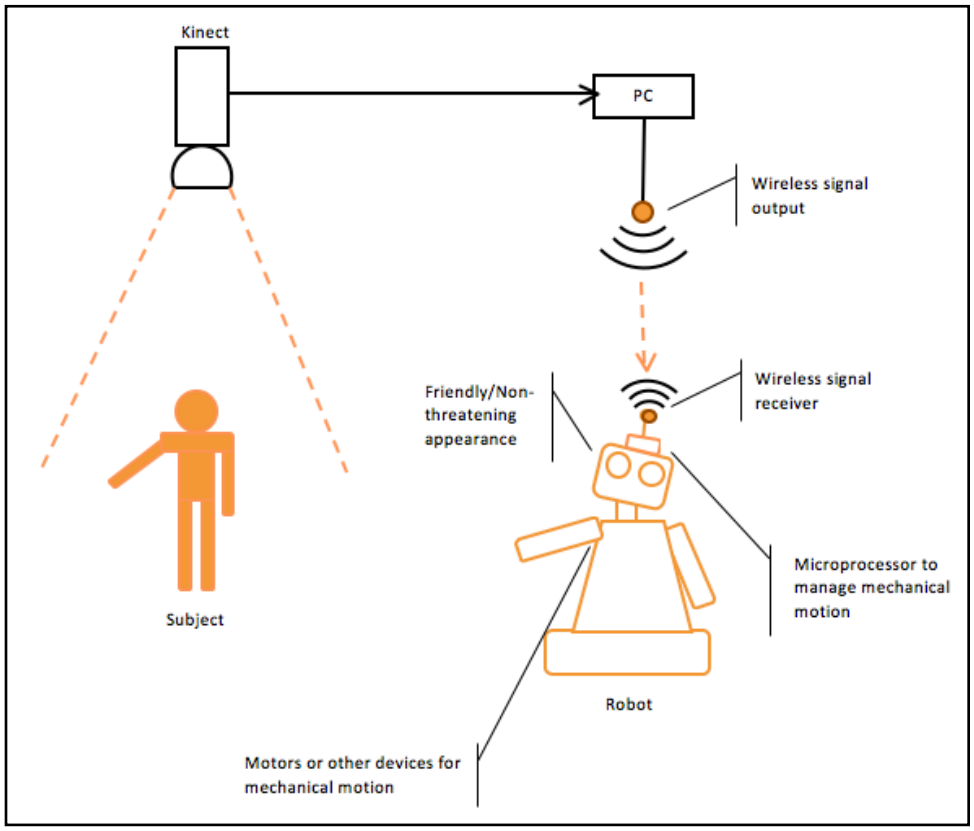
1.2.1 Information

For additional information, the design team can be contacted through Project Leader Kristine Dietrich (dietrich.kristine@gmail.com).

2.0 SYSTEM SUMMARY

2.0 SYSTEM SUMMARY

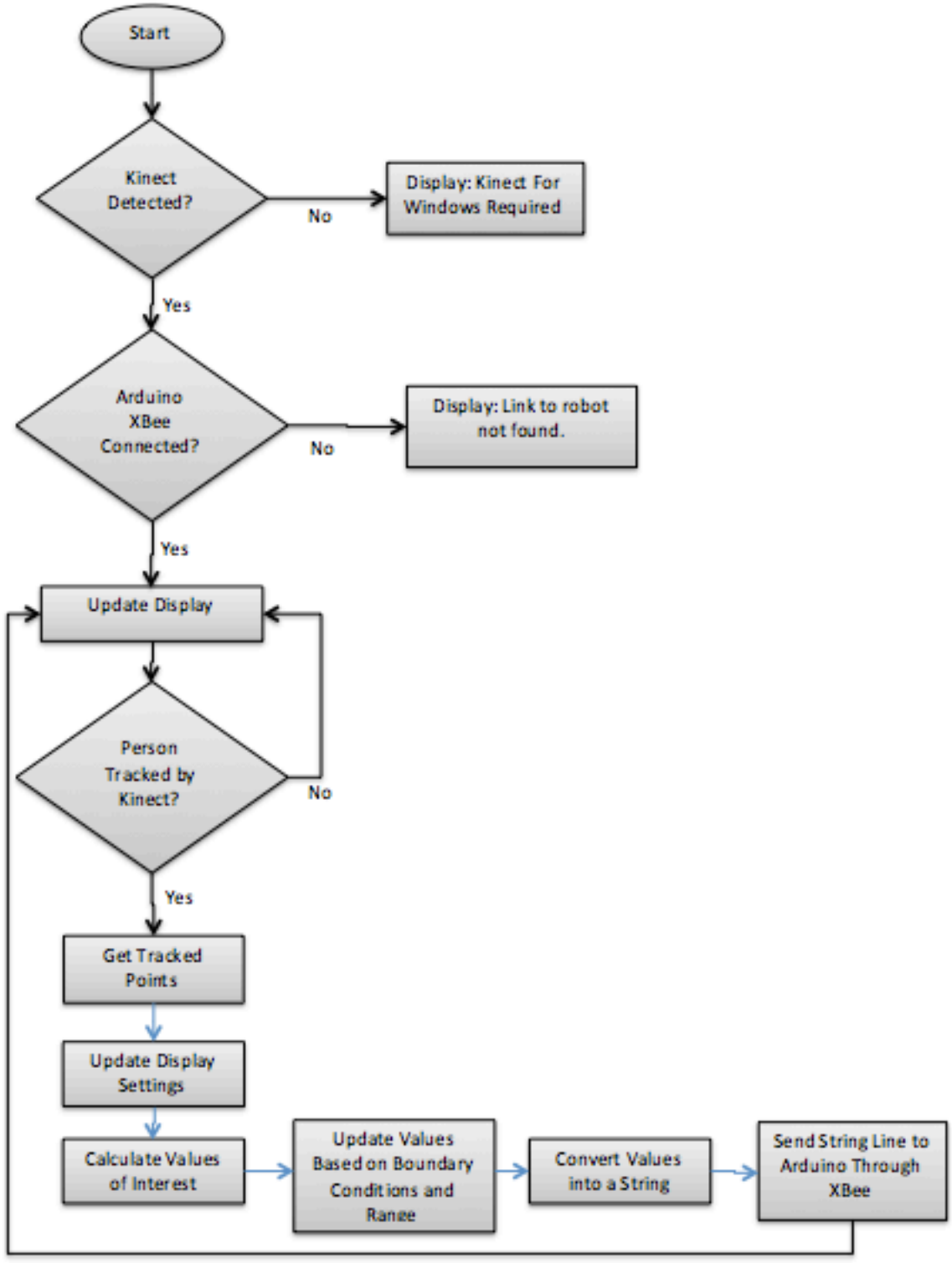
2.1 System Configuration



The system consists of a robot that interfaces with a user though a PC with a Microsoft Kinect.

2.2 Function Flows

The below diagram flow of the loading, connection test, and running of the Social Robotics program.



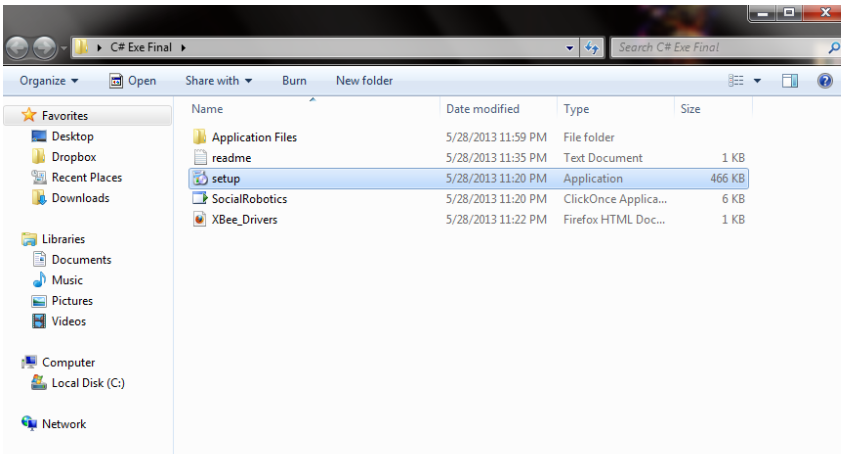
3.0 GETTING STARTED

3.0 GETTING STARTED

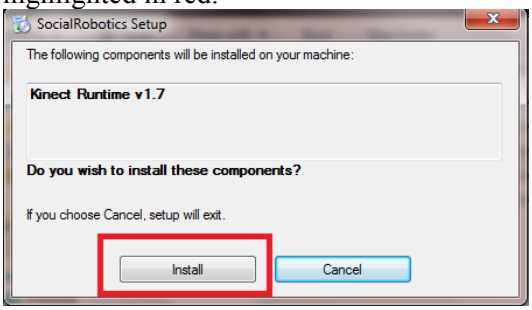
Downloading, installing, setup, use, and trouble shooting of the Social Robot software and robot interface.

3.1 Downloading the Social Robot Program

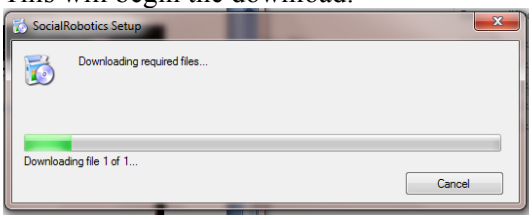
This installation requires an internet connection. Begin by copying the folder "C# Exe Final" to your desktop. Open this folder and click setup to begin the installation process.



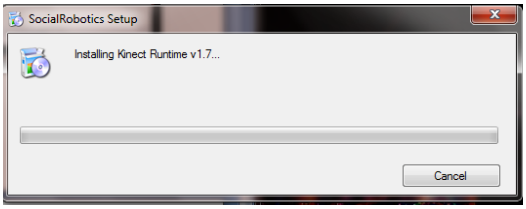
The following window will be brought up asking if you wish to install Kinect Runtime v1.7. Chose Install highlighted in red.



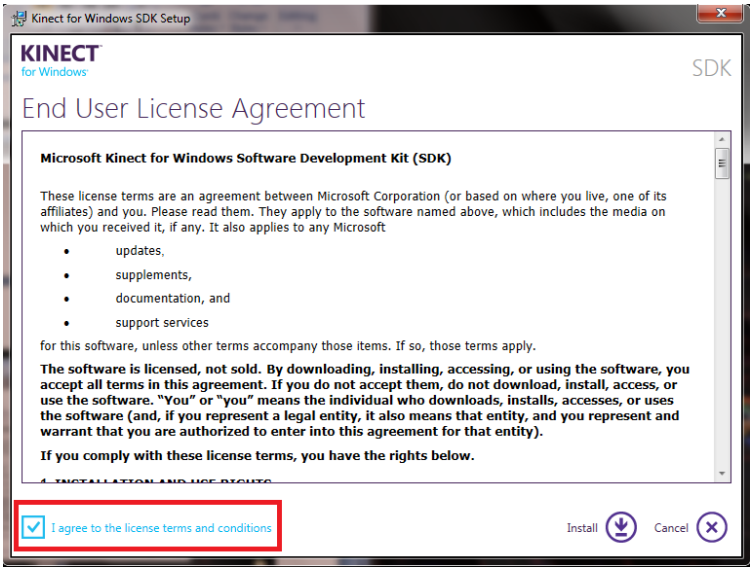
This will begin the download.



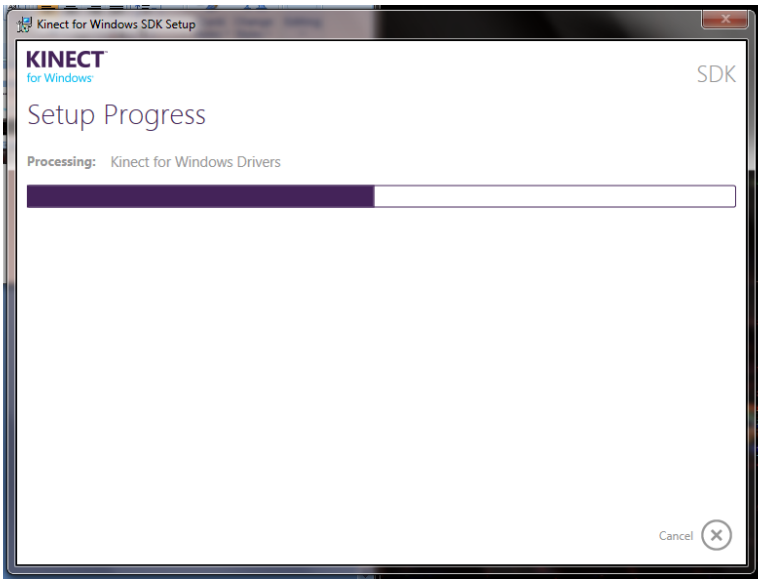
And automatically begin the install process.



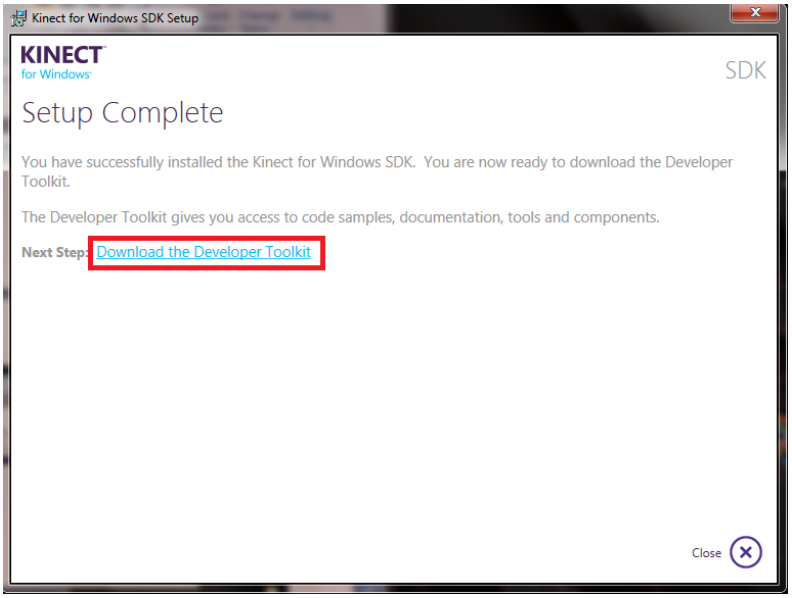
Once this is completed the following screen will be brought up. Showing the End User License Agreement. To continue you must agree to the license terms and conditions. To do this check the box highlighted in red and hit the install button on the right.



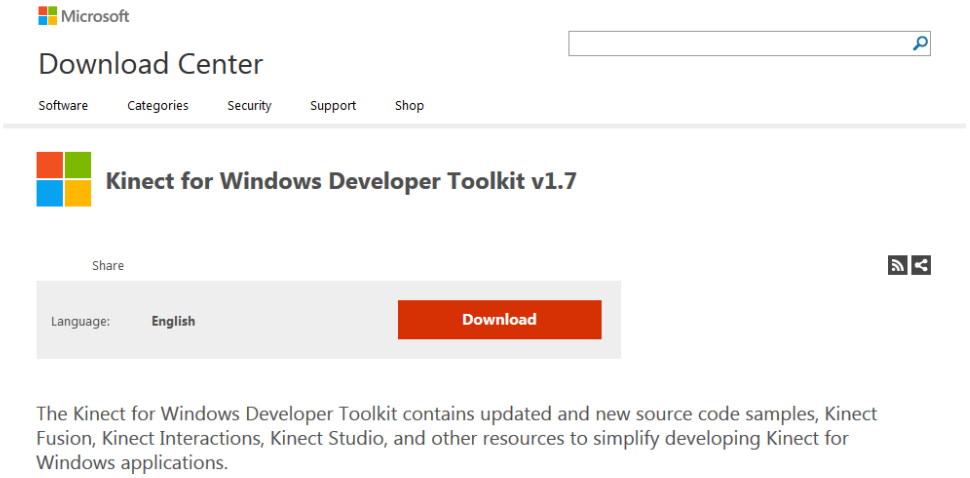
Once you agree to the terms the SDK setup will begin



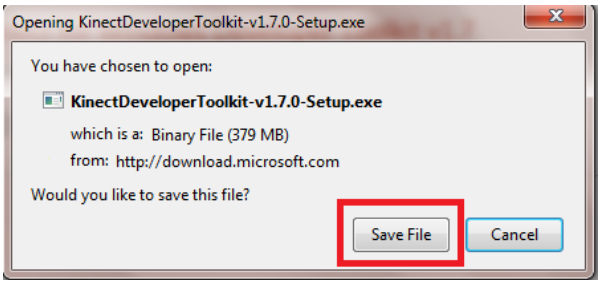
Once the setup is complete you must next download the Developer Tool Kit by clicking on the link.

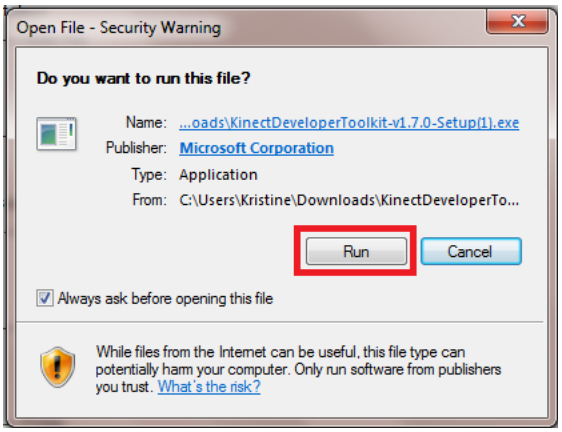


This will open a browser window and bring you to the following page. Click the download button.

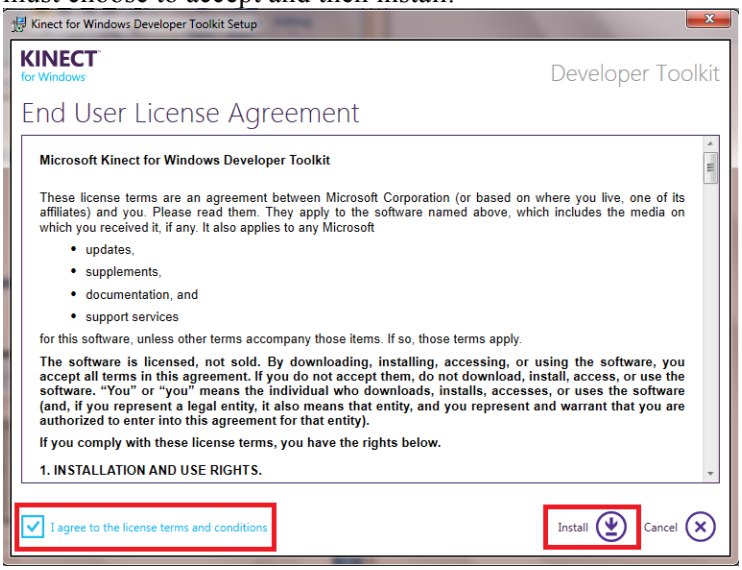


This will bring up the following window. Chose Save File then Run

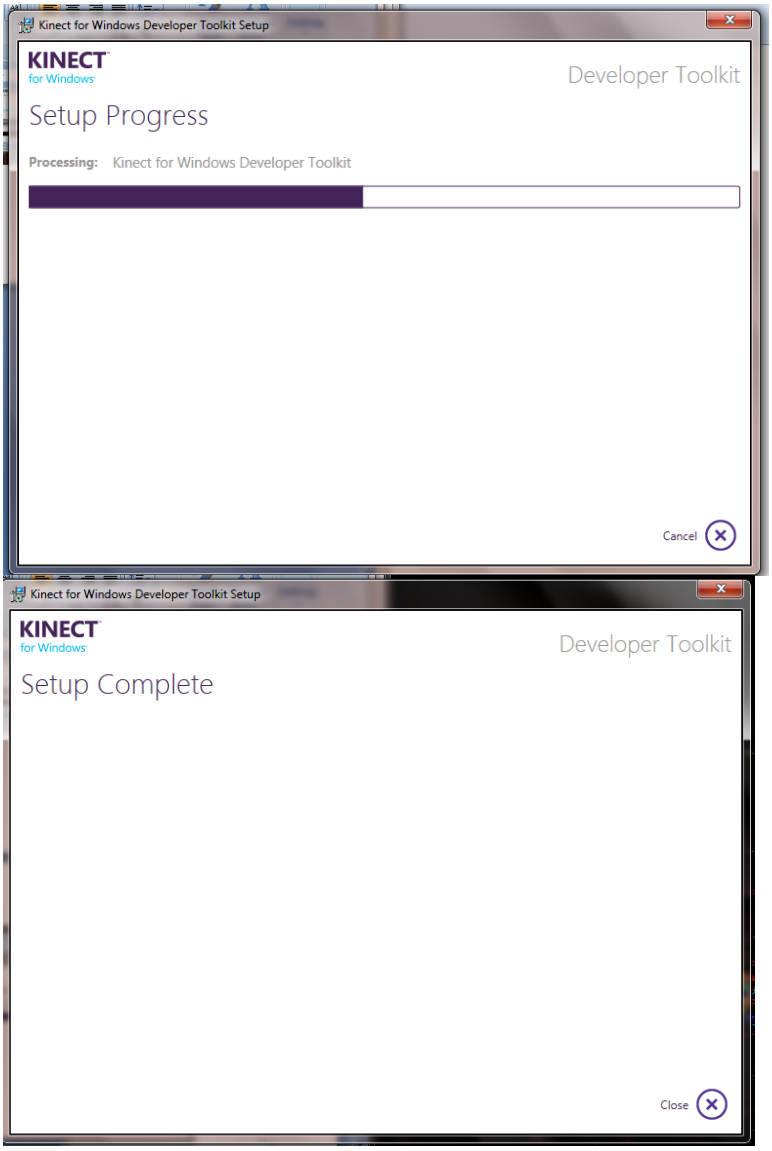




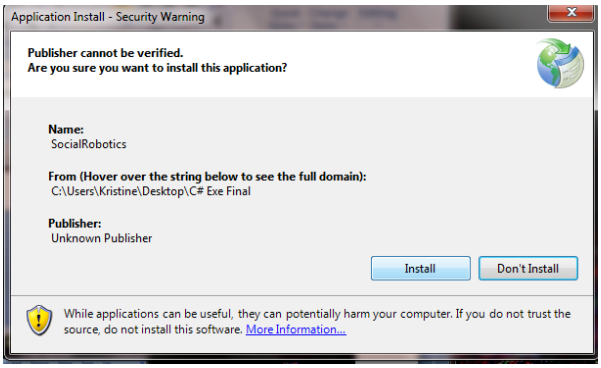
This will bring up the Kinect for Windows Developer Toolkit Setup End User License Agreement. You must choose to accept and then install.



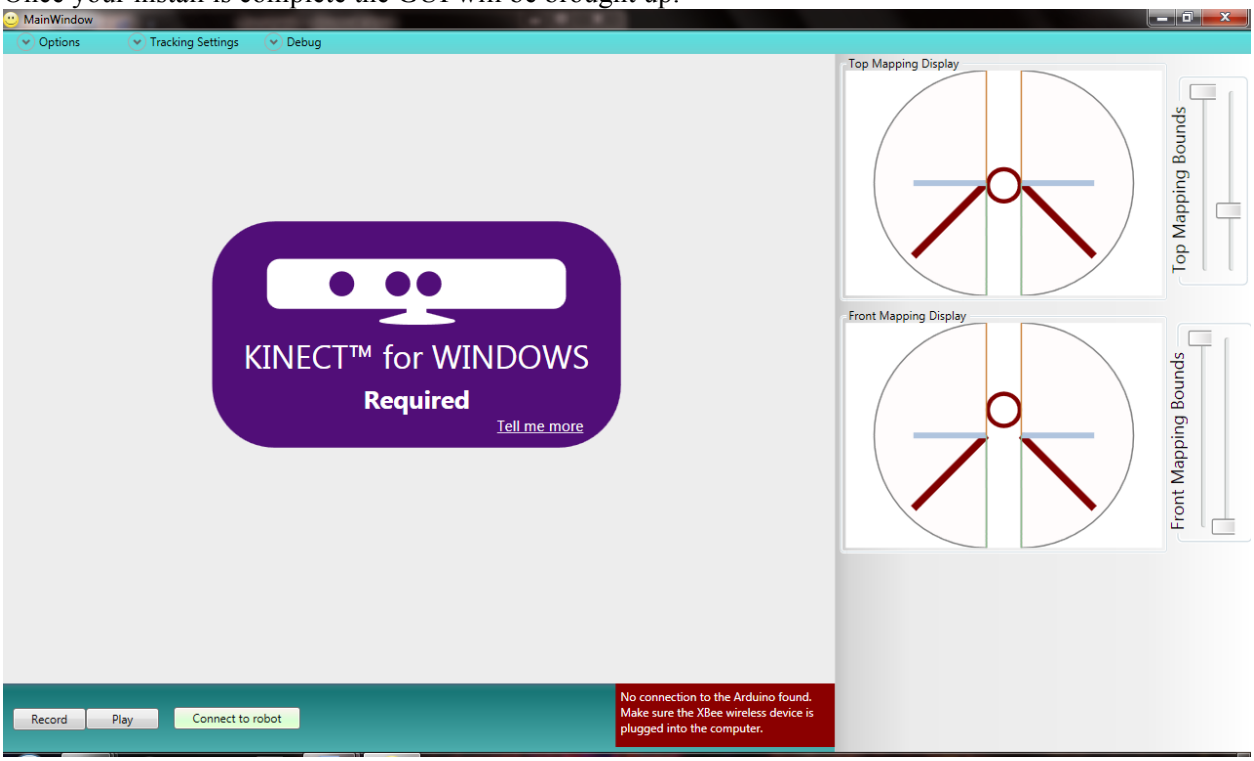
This will begin the Setup Process



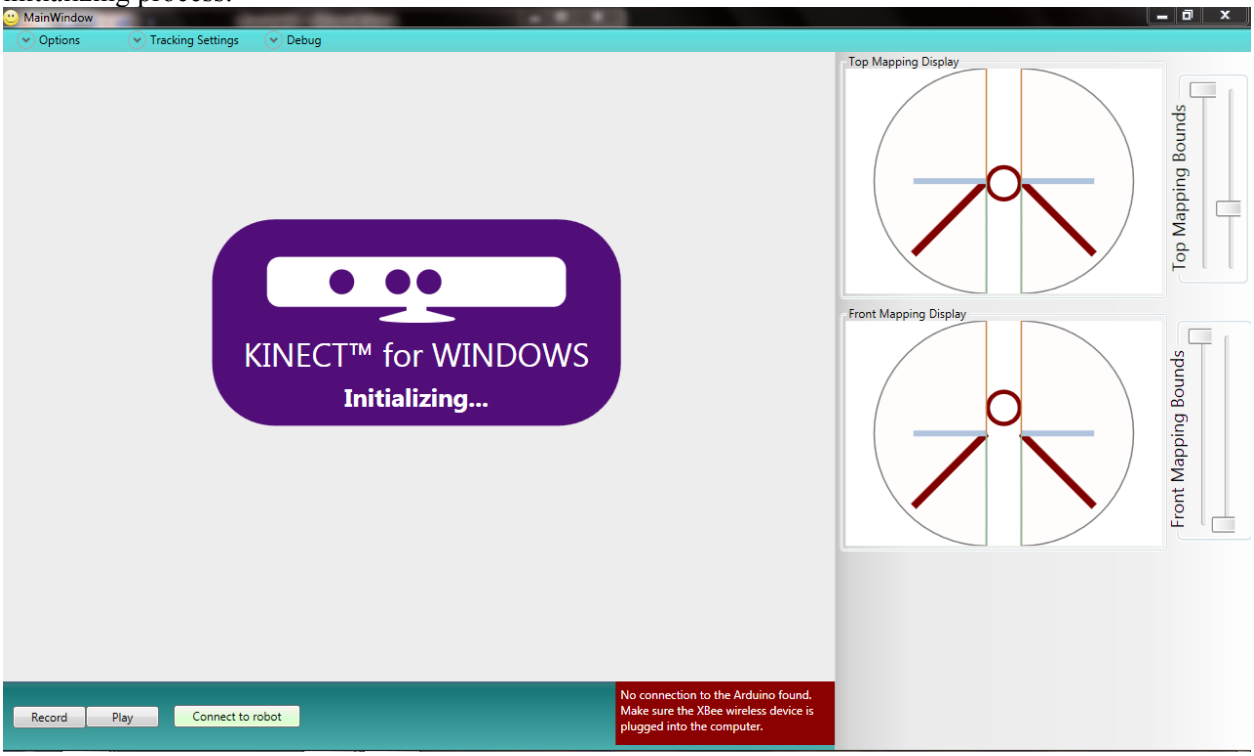
When the setup is complete you will see an Application Install Security Warning, chose to install the program. This is the Social Robotics Program.



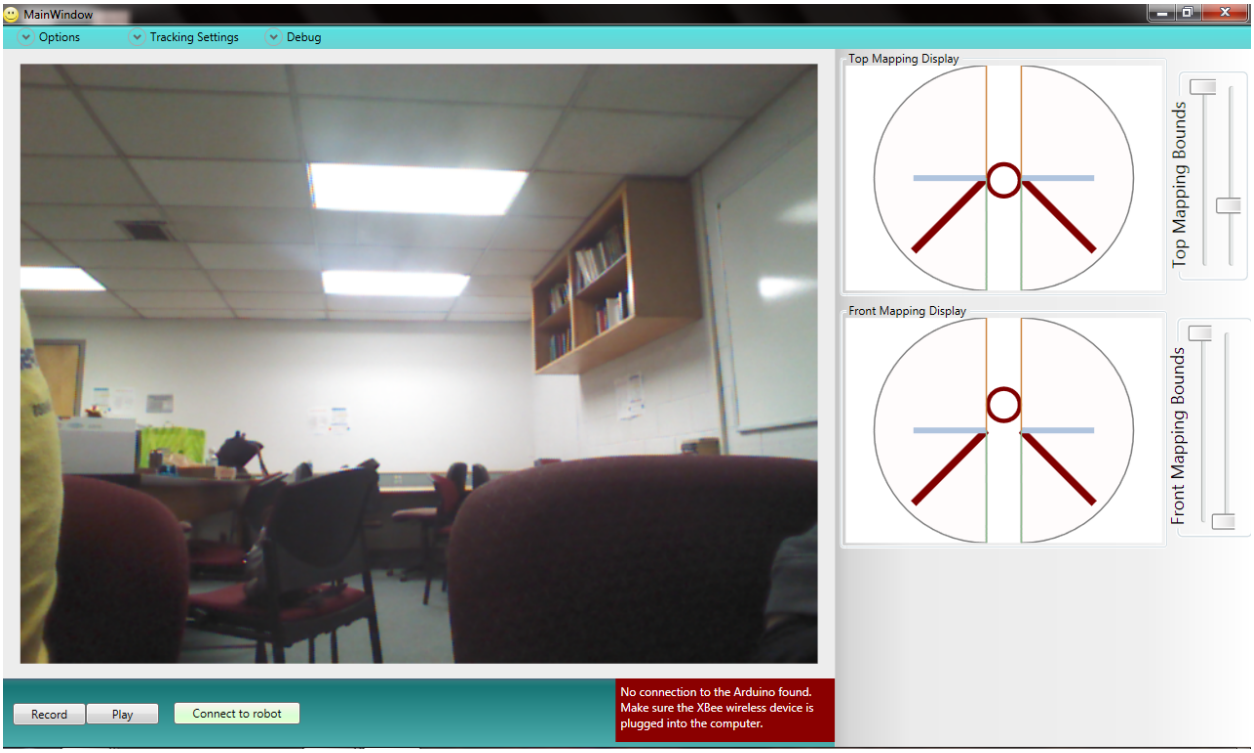
Once your install is complete the GUI will be brought up.



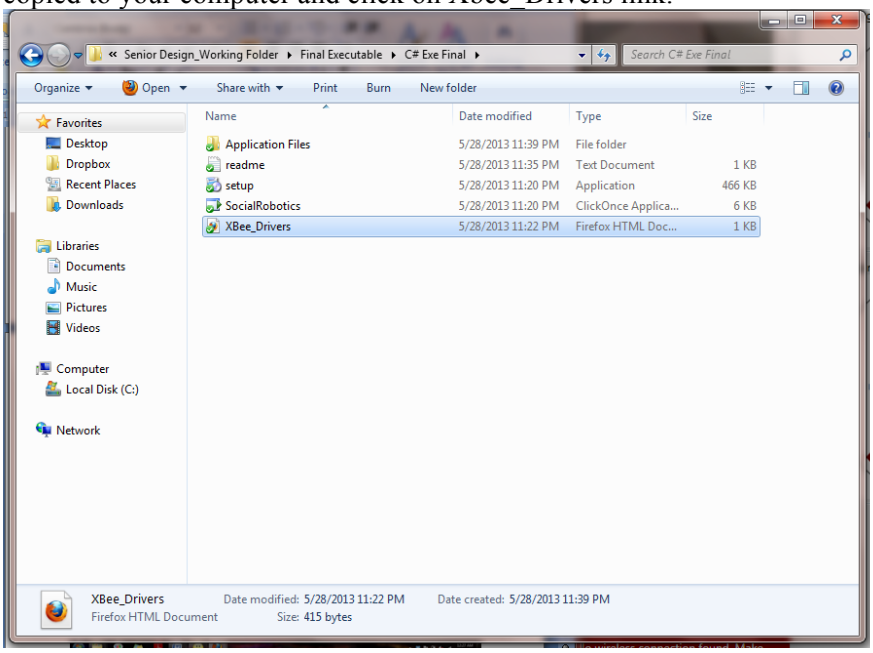
Plug in the Kinect for Windows into a 120V outlet and the usb into the computer. This will begin the initializing process.



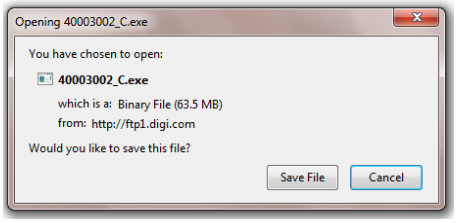
The Kinect is now connected.



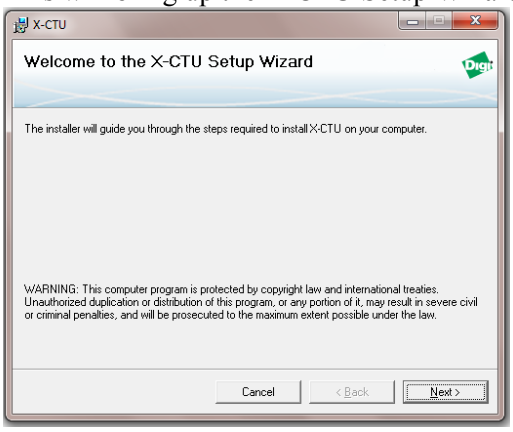
Now you must install the Xbee drivers. To do this you must go back to the C# Exe Final folder you copied to your computer and click on Xbee_Drivers link.



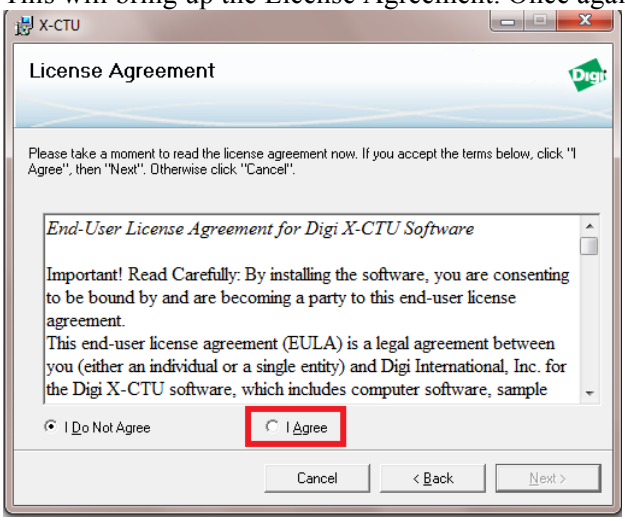
Chose save file



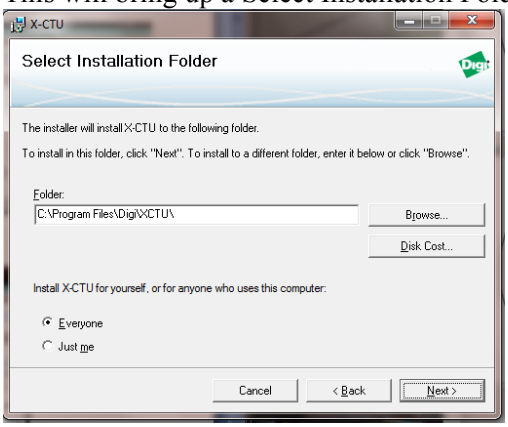
This will bring up the X-CTU Setup Wizard. Chose next.



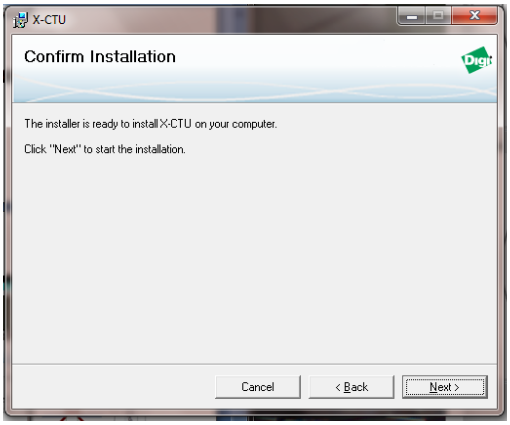
This will bring up the License Agreement. Once again you must chose to agree and click next.



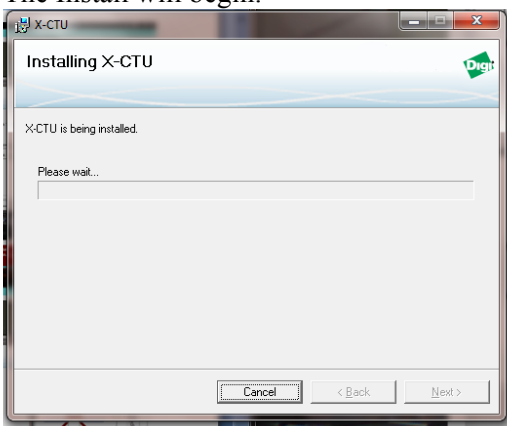
This will bring up a Select Installation Folder. Chose Next.



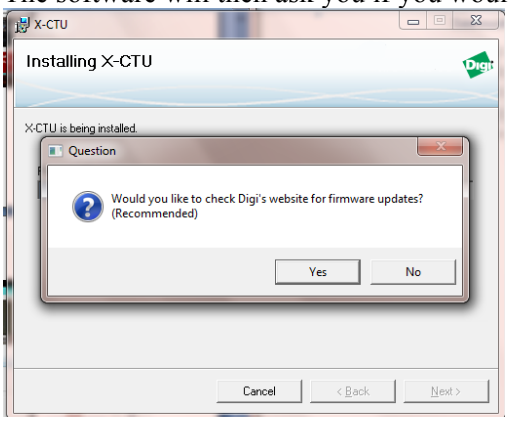
The confim installation window will open . Choose Next.



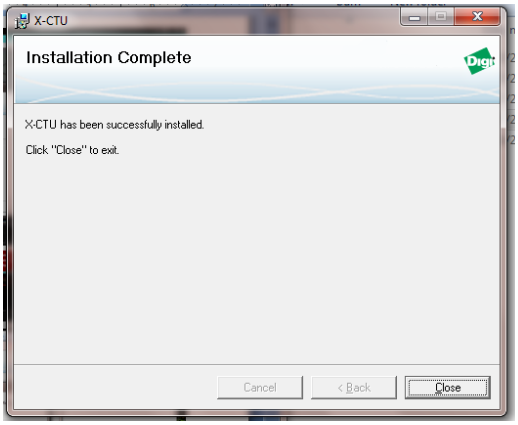
The Install will begin.



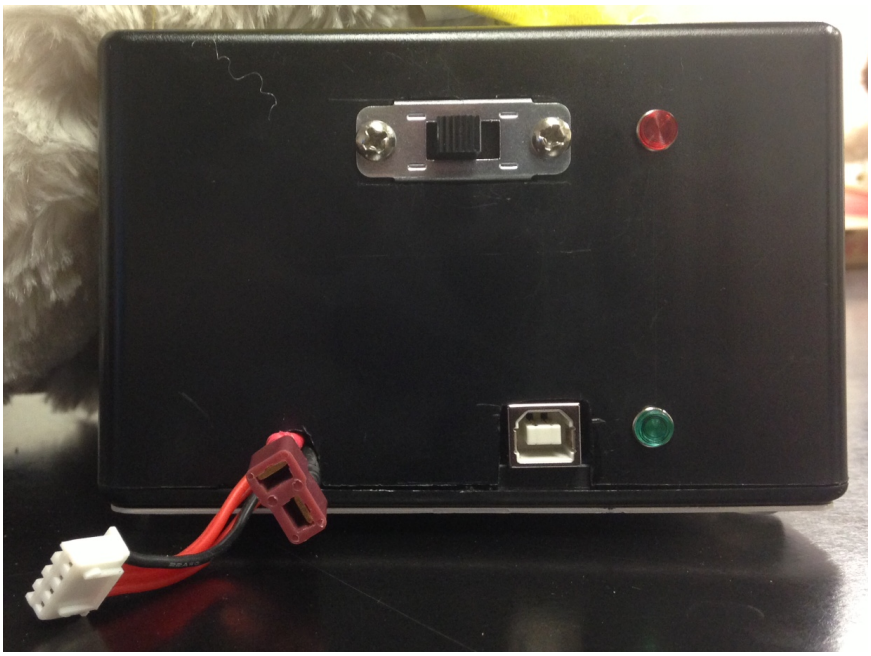
The software will then ask you if you would like to check for updates. Chose Yes.



The system will then check for updates to the drivers. Once this is complete the install will finish.


X-CTU
Installation Complete
X-CTU has been successfully installed.
Click "Close" to exit.
Cancel
< Back
Close

3.2 About the Electrical Box



This picture shows one side of the electrical box underneath your Social Robot. There are many features that are necessary for using your robot.

The most important part is the switch in the top middle of the box. This switch has two settings: “Charge/Off” and “On.” When the robot is not in use, make sure the switch is in the far left, or “Charge/Off” position. This position also allows for charging of the battery. In order to turn the robot on, the switch must be pushed to the far right, or “On” position.

The red LED located next to the switch is the power indicator. When the switch is turned “On,” the LED should be glowing, and it should turn off when the switch is in the “Charge/Off” position.

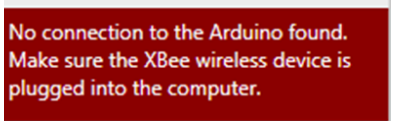
The wires coming from the bottom left of side of the electrical box are used for charging of the battery. Before connecting those wires to the battery charger, make sure the switch is in the “Charge/Off” position (far left). The battery charger manual accompanies this manual.

The other connection that can be made from outside the electrical box is for programming of the microprocessor of the robot. Unless you are an experienced programmer, do not use this connection.

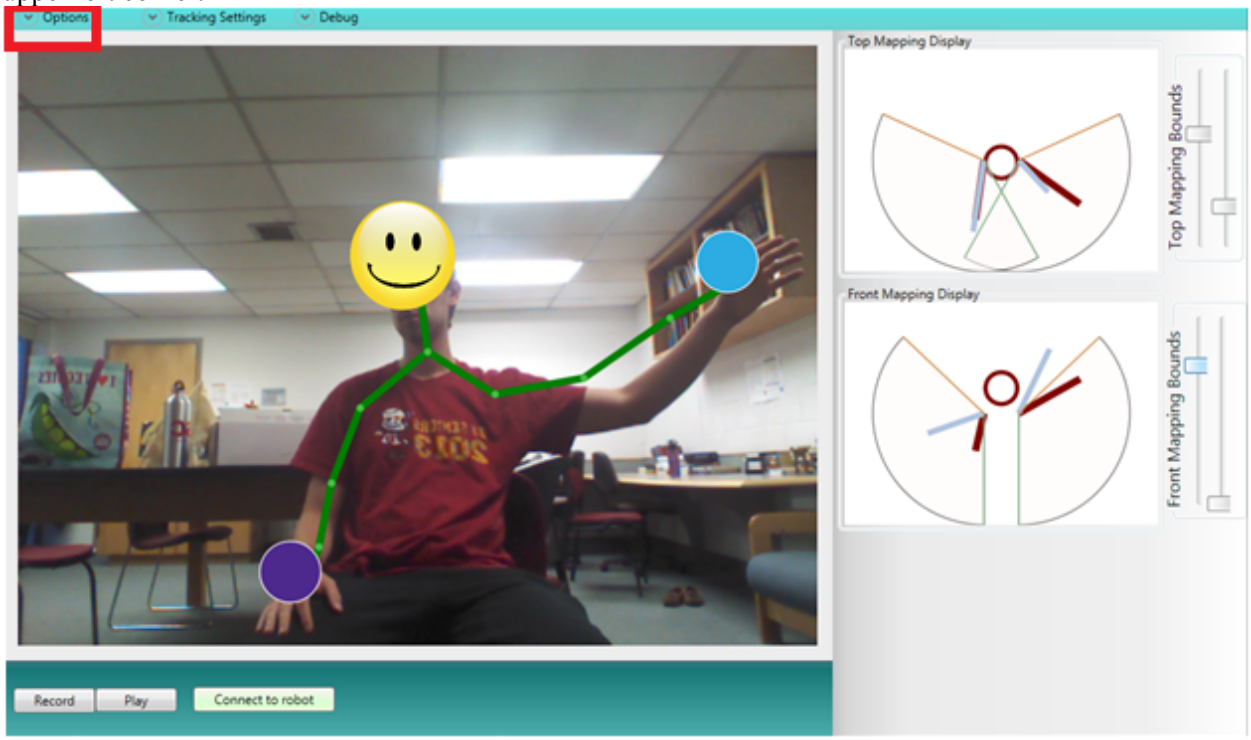
The green LED located next to the microprocessor connection shows whether or not the PC and microprocessor are communicating. If the green light is illuminated, they are connected. If the light is not illuminated, the PC is not recognizing the robot and some troubleshooting must be done.

3.3 Using the GUI

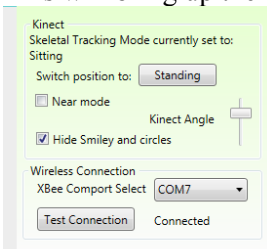
Now that everything has been downloaded and installed, Plug in the Xbee. Now we will begin using the GUI. The following error message may appear.



If the Xbee is indeed plugged in you must check the Com Port. To do this go tho the options tab in the upper left corner.



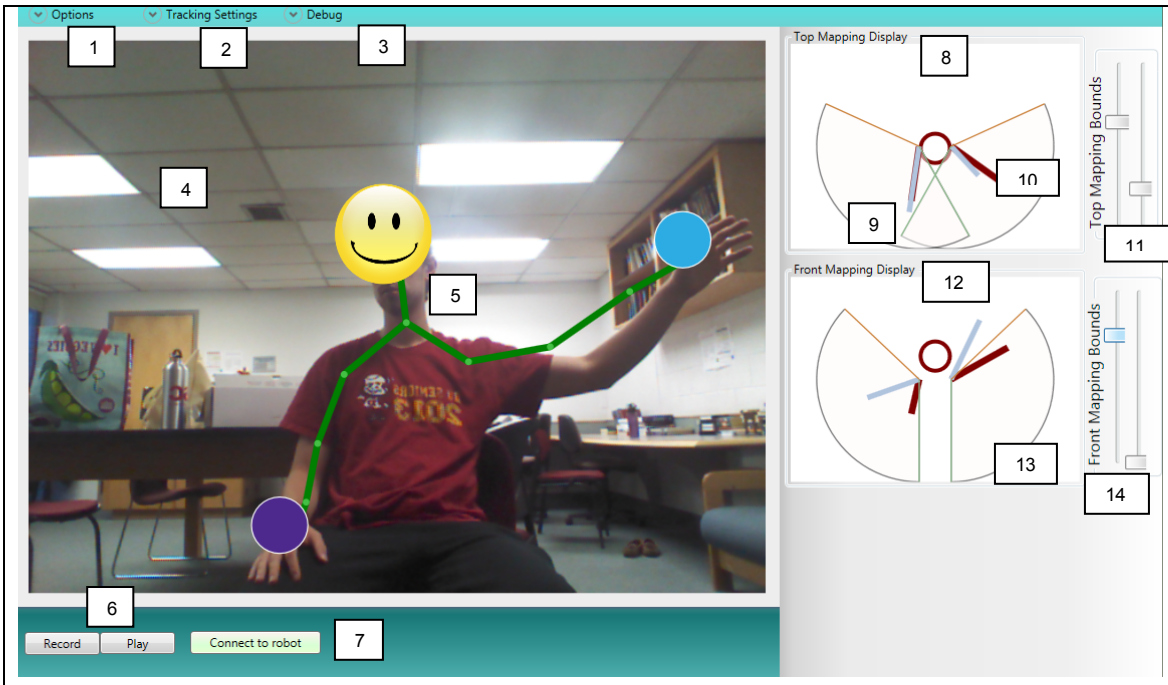
This will bring up the following tool set.



Select the correct comport. You can test the connection using the test connection button at the bottom. If there is still no connection make sure the robot is turned on and the red power light is on. Press connect to robot in the bottom of the GUI screen. When the connection is made the button will switch to disconnect from robot and the green LED on the robot will be on. Step away from the robot and begin use!

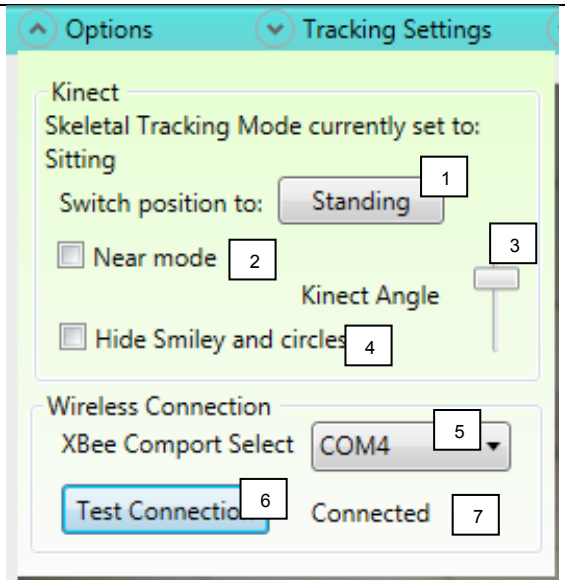
3.4 Advanced GUI Settings

The GUI includes some advanced settings. First we will get a more in depth look at the GUI itself. Here is a diagram of the GUI interface with each number adding descriptions.



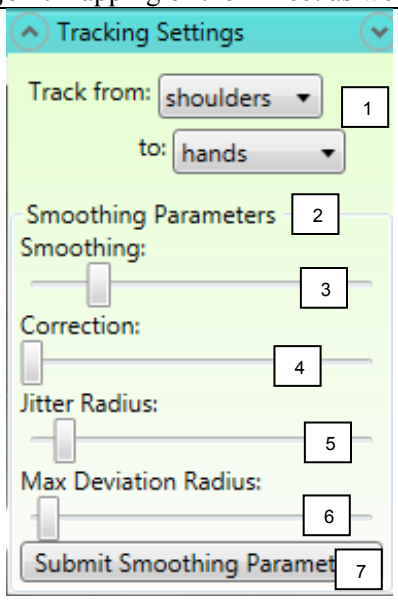
1. Options pull down – Allows for setting Kinect and Wireless connection settings
2. Tracking Settings – Settings for what joints to track and the tracking smoothing parameters
3. Debug pull down – Allows users to see and input additional behind-the-scenes data
4. Main window display - Shows Kinect camera view
5. Kinect tracked skeleton – Shows the joints tracked by the Kinect in an overlay
6. Buttons for recording and playing movement sequences
7. Button for connecting/disconnecting from the Arduino
8. Visual display of top down view of users arm motion and the robots corresponding response
9. Light blue vectors represent robots arm motions
10. Dark red vectors represent users arm motions
11. Sliders for setting the horizontal range of user motion that is mapped to the robots full range
12. Visual display of front view of users arm motion and the robots corresponding response
13. Visual representation of mapping range shown by arcs
14. Sliders for setting the vertical range of user motion that is mapped to the robots full range

Here shows a more indepth view of the Options menu which can be found in the upper left hand corner of the GUI.



1. Set the Kinect tracking mode to sitting or standing
2. Set Kinect tracking distance setting to default or near mode
3. Adjust the vertical Kinect Angle
4. Set the visibility of the tracked smiley face and hand circles
5. Select the comport for the XBee wireless connection to the Arduino
6. Button for testing the wireless connection status
7. Status of wireless connection

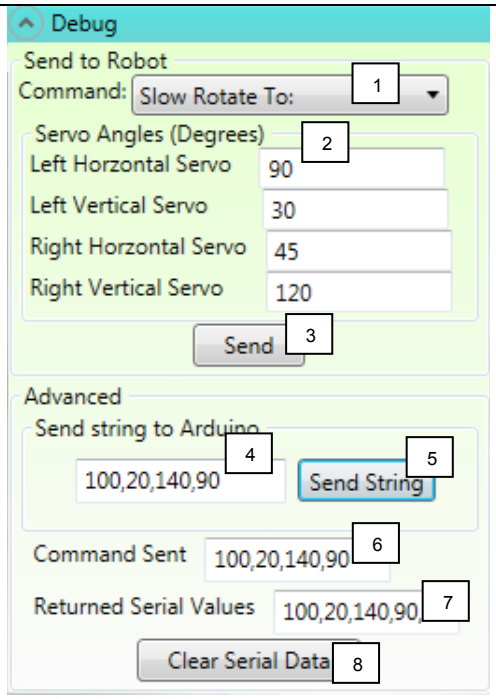
The next tab over is the Tracking Settings which is detailed below. In this section the user has the ability to adjust the joint mapping of the Kinect as well as a variety of other settings.



1. Select which joints the program should track as endpoints of the vectors to use for calculations
2. Set the smoothing parameters for how the Kinect smooths the tracked joints motions
3. Smoothing value: Sliding to the right corresponds to more smoothing. Increased smoothing leads to increased latency.
4. Correction value: Higher values correct faster, but is less smooth
5. Jitter Radius: Determines how aggressively to remove jitter from the raw data.
6. Max Deviation Radius: Determines how aggressively to remove jitter from the

raw data.

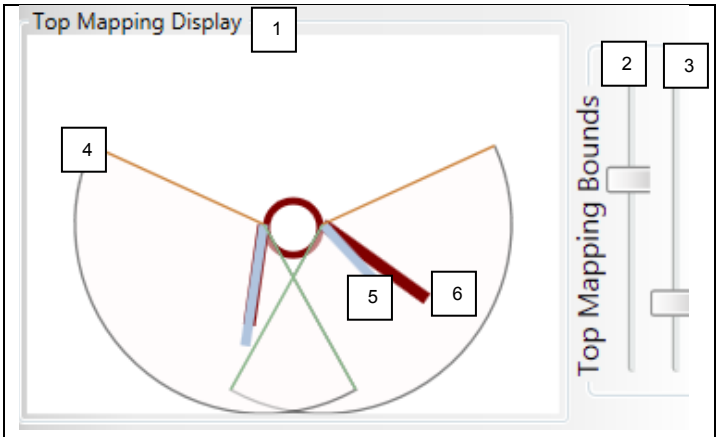
The next tab over is the Debug menu.



1. Select a command to send to the Arduino
2. User input angles to send to the servos (naming convention here represents servos from the robots perspective)
3. Button to send data from above
4. Manually enter data string to be sent to the Arduino (in form 'a,b,c,d,e,' where a through e are integers)
5. Send the string immediately preceding this button to the Arduino
6. Shows the last command sent to the Arduino
7. Shows returned signals when using command 100
8. Clears the serial port of any current data

The final part of the GUI is the angle mapping. This angle mapping defines how the arm motions of the robot relate to the arm motions of the user. The GUI has two displays for visualizing the users arm motions and the corresponding robot arm motions. These displays represent a top down, and front on projection. The user's arm vectors are dark red, and are based on joints selected in the *Tracking Settings* menu. The robot arms are represented in light blue vectors.

The mapping between the user and robot arms is represented visually by two arc sections. These arcs represent the range of motion for each arm that corresponds to a full range of the robots arm motions. The sliders to the right of the projected display can be used to adjust this mapping range.



1. Lable indicating front view of arms
2. Slider for controlling upper bounds of user arm motion mapping
3. Slider for controlling lower bounds of user arm motion mapping
4. Visual display cue of upper range mapping
5. Light blue: Robots corresponding arm motion
6. Dark red: Users actual arm position

4.0 TROUBLESHOOTING

This section will help you if you run into any problems while using your social robot.

4.1 Common Problems

Connection Problems

a. Check that XBee wireless device is plugged in to your USB port.

b. Make sure the robot is turned on (red LED should be illuminated).

c. In the 'Options' menu, make sure the correct COM port is selected.

   i. If the XBee is plugged in after the program has been started, the connection will only work if the XBee connects to COM4. To be safe, connect the XBee before loading the program.

d. Use 'Test Connection' button in the 'Options' menu to check the connection status.

   i. If you are connected, the green LED on your robot should be illuminated **and** it should say 'Connected' on the GUI.

Kinect Issues

a. Make sure your Kinect for Windows is plugged in to a wall outlet. When you do this, a power light on the cable should be illuminated.

b. Make sure your Kinect for Windows is plugged in to your USB port. When you do this, a green light is visible on the front of the Kinect.

c. You must be in the acceptable range to be tracked by the Kinect. There are two field of view modes available on the 'Options' menu of the GUI: default and near. For each of these, there are different acceptable distances.

   i. Default: 2.6' - 13.1'

   ii. Near: 1.3' - 9.8'

   Select the appropriate mode for your needs.

Program Starting Issues

a. Once you have installed the program to a certain place on your computer, it cannot be moved.

   i. If it has been moved, first uninstall and then reinstall. To uninstall, open the Start menu on your computer.

      Click ‘Control Panel’

      Click ‘Uninstall A Program’

      Sort to find ‘SocialRoboticsBear’ and select it

      Click ‘Uninstall/Change’ at the top of the window

      Click ‘Ok’ to remove the application

      Once uninstall is done, you are free to reinstall the program

b. Make sure that you still have the Microsoft Kinect SDK installed.

Movement Issues

a. Ensure that your battery is charged. If it is not charged, the red light on your robot will not be illuminated.

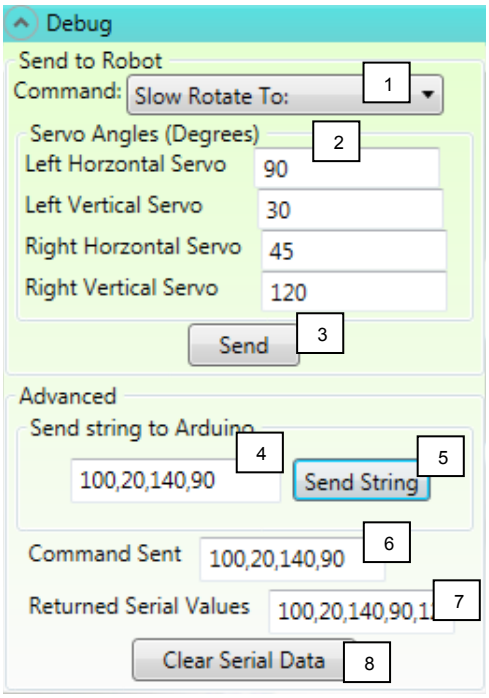
b. Make sure all connections have been made. Check the Connection Problems Troubleshooting section.

c. With device in ‘Off’ mode, gently move arms in all directions to ensure that mechanical system is intact.

d. If problems persist, some internal stuffing may be blocking movement. To resolve this, remove outer covering and move the stuffing away from all moving parts.

4.2 Using the Debug Menu

In the GUI, there is a 'Debug' menu that can be useful if you are experiencing any problems. Below is a picture of the menu and detailed description of how to use each functionality.



1. *Select a command to send to the Arduino*

    a. The range of motion selections can be useful for debugging robot movement issues

        i. Default - rotates all servos to the angles specified with (2)
        ii. Slow Rotate To - does the same as default but very slowly
        iii. Return the command 100 - will resend the input to the returned serial values text box
        iv. Slow Range of Motion - moves servos over their full range slowly
        v. Fast Range of Motion - moves servos over their full range quickly
        vi. Neutral, Arms Down - returns servos to their neutral location

2. *User input angles to send to the servos (naming convention here represents servos from the robots perspective)*

    a. In options i, ii, and iii from (1), these values are specified as to what the desired servo rotation will be

3. *Button to send data from above*

a. Submits command and input angles. Results in a movement of the servos.

4. *Manually enter data string to be sent to the Arduino (in form 'a,b,c,d,e,' where a through e are integers)*

   a. More advanced but similar action as above. First integer represents the command and other four are servo angles. The following are the appropriate first integer values and their command description.

Command #	Description
0	Rotates servos to angles defined by degree number from next 4 integers (order: 'Left horizontal servo', 'Left vertical servo', 'Right horizontal servo', 'Right vertical servo')
1	Robot transitions to its neutral arms down position
20	Same as command '0', but where it rotates the servos very slowly to their final value defined by following 4 integers
100	Same as command '0' except the Arduino resends the command it received back over the serial port connection (for the handshake test)
101	Turns on LED verifying connection to C# program on computer
104	Slowly rotates the servos over their range of motion
105	Quickly rotates the servos over their range of motion

5. *Send the string immediately preceding this button to the Arduino*

   a. Sends the values input from (4)

6. *Shows the last command sent to the Arduino*

   a. Displays the last sequence which was sent to the Arduino. May be either from when the program was running autonomously or a sequence sent in by the user.

7. *Shows returned signals when using command 100*

   a. The returned value from the Arduino imitating the original command sent. Tells you that the connection is completely working if all of the values are the same.

8. *Clears the serial port of any current data*

   a. Used to reset the serial connection in case you are receiving faulty or unusual data.